SIMULATION OF AN INTERNATIONAL STANDARD TRANSPORT PROTOCOL

Kevin Mills - Marnie Wheatley - Richard Colella
National Bureau of Standards

ABSTRACT

This paper describes a simulation model of an international standard transport protocol. The model has been applied to several simulation experiments in support of the Institute for Computer Sciences and Technology protocol performance research program. The results of a simulation experiment to evaluate alternate acknowledgement strategies are reported. A discussion of model validation is included and a short description of future work is provided.

I. INTRODUCTION

There are two main concerns common to users, implementors, and designers of computer communication protocols. First, the protocols must operate correctly. Second, the protocols must operate economically. The Institute for Computer Sciences and Technology (ICST) at the National Bureau of Standards (NBS) has successfully established a program of protocol correctness testing [1,2]. The ICST is now beginning to address the second common concern - economical operation of protocols, or protocol performance. Toward this end, the ICST has initiated a program that will examine performance issues associated with International Standards Organization (ISO) open systems interconnection (OSI) protocols [3]. The first protocol under consideration is ISO class 4 transport [11, 12].

An important part of the ICST program of protocol performance research is modeling existing and proposed protocol mechanisms. This paper describes a simulation model of the ISO class 4 transport protocol based upon the NBS formal description [4, 5]. The motives for developing the model are outlined and an overview of the model is given. The validation of the model is described. The performance metrics that can be obtained from the model are discussed and some modeling results are presented. Finally, a section is included that describes future work to be conducted by the ICST as the research program continues. Readers requiring a more general understanding of the simulation of computer communications systems should see Sauer and MacNair [6].

The model described in this paper has been used in support of a joint experiment program between the NBS and COMSAT. Model results were used to plan live experiments with the transport protocol over a satellite channel. The results of the experiment planning are described in a separate report [10]. Several of the planned experiments have been conducted over a satellite channel and the live experiment results will be reported jointly by the NBS and COMSAT.

II. MOTIVATION FOR USING A SIMULATION MODEL

Since designing, implementing, and using a simulation model of a complex system can require substantial manpower and computer time, a strong motive is required for relying upon such a model. The four reasons behind our decision to develop a simulation model of the ISO class 4 transport protocol are presented below.

First, our program of protocol performance evaluation calls for the design and execution of live experiments using various network technologies. Running each live experiment requires substantial expense in terms of software development, use of network services, and experiment management time. Execution of simulated experiments, however, requires minimal expense. Thus, a wider range of experiments can be performed via simulation, reducing the number of live experiments to a truly significant set.

Second, an objective of our research is the evaluation of performance improvements that might be realized through the use of alternate mechanisms for various protocol functions such as error control, expedited data, and acknowledgement. Specifying, designing, implementing, and testing such alternate mechanisms within an existing protocol implementation is expensive; therefore, some knowledge of the potential performance improvements from each proposed mechanism is needed before resources are allocated to build and test the protocol software. A simulation model is a suitable tool to gain such knowledge. The model allows us to reject those mechanisms that do not improve performance and to refine those that remain, eliminating undesirable properties before implementation.

Third, we preferred a simulation model to an analytical model. While analytical models are easier to create, use, and understand than simulation models, they are limited in the amount of system detail that can be represented. This limitation results from the high level of abstraction required to produce mathematically tractable equations. Also, even though an exact mathematical expression results from an analytical model, its computational accuracy is constrained by the numerical methods used.

Fourth, simulation models permit the collection of performance measures that are difficult or impractical to collect during live experiments. Also, the measures available from simulation models can be much more detailed than those obtainable using analytical models.

III. MODEL OVERVIEW

The model described in this paper, as shown in Figure 1, consists of three major modules: 1) user, 2) transport, and 3) network. The user module produces and consumes traffic for the simulated transport module. The traffic is generated in accordance with model input parameters provided for each simulation run.

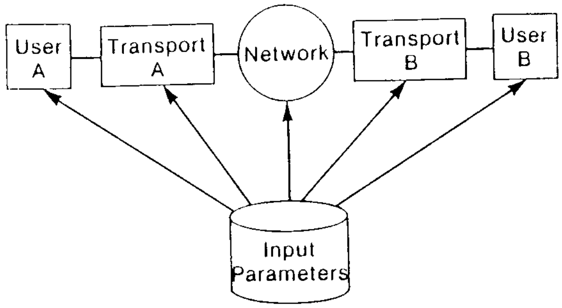

Figure 1. Transport Simulation Model—Basic Structure

The transport module simulates the data transfer phase of ISO class 4 transport for both normal and expedited data. Control is provided over transport parameters such as retransmission timers, window timer, window size, maximum message size, maximum acknowledgement delay, degree of acknowledgement withholding, and amount of receiver and sender buffer space [13, 14]. The module simulates class 4 transport mechanisms for error detection and recovery, flow control, segmenting, and reassembly. The transport module reads simulated data from and delivers simulated data to the user module(s) it serves in compliance with the specific interface rules modeled for each user. The transport module also simulates exchange of data across the network interface in conformance with the rules of the specific network type being simulated.

The network module simulates the characteristics of a network, specified as a part of the model input parameters. Characteristics simulated include data rate of the subnetwork access link, network propagation delay, network flow control restrictions, and network error properties (probability of loss, misordering, damage, and duplication).

A. INTERNAL STRUCTURE

Figure 2 illustrates the internal structure of the simulated user, transport, and network modules. The user module controls the generation and consumption of normal and expedited interface data units (IDUs and XIDUs, respectively). The user also marks transport service data units (TSDUs) within the stream of generated IDUs, and enforces the sequencing of IDUs and XIDUs across the flow controlled user-transport interface.

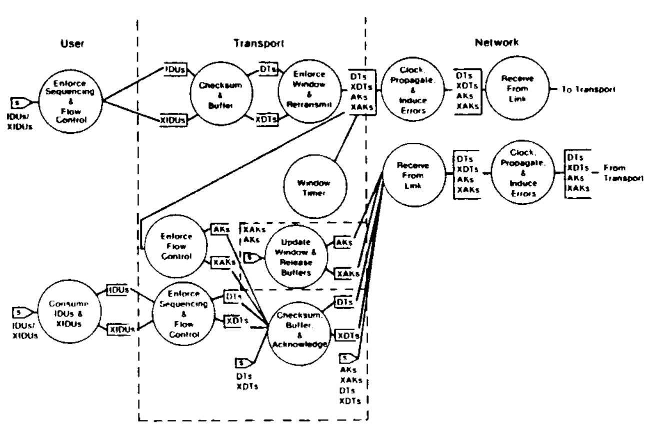

Figure 2. Transport Simulation Model—Internal Structure

The simulated transport module is composed of three main threads of processing: 1) send data, 2) receive data, and 3) receive acknowledgements. The send data thread includes two steps. Step one of the send data thread consists of receiving IDUs and XIDUs from the user, formatting normal and expedited transport protocol data units (DTs and XDTs, respectively), computing checksums for the DTs and XDTs, and storing them into the transmit buffer. Step two of the send data thread enforces the various flow control restrictions placed on DTs and XDTs (including window size, network interface flow control, and expedited and normal data sequencing) and controls the retransmission of DTs and XDTs.

The receive data thread of the simulated transport module includes four steps. Step one consists of processing the received DTs and XDTs. The checksum for each message is tested; "failing" messages are discarded and "passing" messages are merged into the appropriate receive buffer. Any acknowledgements required are generated. Step two controls the creation and forwarding of IDUs and XIDUs to the receiving user. The IDUs and XIDUs are subject to the sequencing controls applied to the DTs and XDTs that compose them and to the interface flow control between the user and transport.

The third and fourth steps of the receive data thread control the transmission of normal and expedited acknowledgements (AKs and XAKs respectively). Step three enforces the network interface flow control required for AKs and XAKs generated in response to received DTs and XDTs.

The fourth step controls the generation of AKs in response to expiration of the window timer. The window timer guarantees an upper bound on the time that can elapse without transmission of an AK. The window timer is restarted each time the simulated transport module generates an AK.

The final thread of processing in the simulated transport module, receive acknowledgement, entails a single step. The checksum for each received AK and XAK is tested and "failing" messages are discarded. "Passing" messages cause update of the appropriate transmission windows and permit release of the appropriate DTs or XDTs that had been held for possible retransmission.

The simulated network module is composed of one processing thread that is used in two directions to give simulation of a full-duplex service.

The network module provides a limited send queue for each transport module so that interface flow control can be regulated. Any messages in the send queue (DTs, XDTs, AKs, and XAKs) are individually clocked onto a subnetwork access link as the link becomes available. The messages propagate and the appropriate errors are introduced. Once a message has propagated to its destination, a link receive service places the message in the appropriate buffer area. If the required buffer area is full, the message is discarded.

B. PROGRAM DETAILS

The model program is written in the C language and runs at the ICST on a VAX-11/780 computer under the VMS operating system. Standard C library functions have been used so the program should be easily portable to a UNIX™ environment. The code consists of about 10,000 source lines.

™UNIX is a trademark of AT&T.

The load module contains a program segment of about 270,000 bytes and a data segment of about 37,000 bytes. Memory for additional data structures is allocated dynamically during program execution. Dynamic memory requirements vary according to the data traffic parameters, the number of transports and users being simulated, and the specific mechanisms being modeled. The model can run as a batch job or interactively. Interactive operation provides access to a range of capabilities for controlling the model that are not available during batch operation.

The performance of the model program can be characterized in several ways, including: 1) simulated events per CPU second, 2) simulated seconds per CPU second, 3) simulated total bits transferred per CPU second, and 4) simulated user bits transferred per CPU second. Table 1 presents this information from twenty-one program runs used to produce the simulation results described later in this paper (Section VI). The two sets of one-for-one data permit comparison of model runs made with and without a floating point accelerator (fpa). The other comparison that can be drawn is between the processing requirements of simulated one-for-one acknowledgement and simulated selective acknowledgement. (See the next section for an explanation of these acknowledgement mechanisms).

C. ALTERNATIVE MECHANISMS

To support our research the model selectively simulates several different acknowledgement strategies and various expedited data mechanisms. The acknowledgement mechanisms simulated include: 1) one-for-one, 2) one-for-N, and 3) selective. These acknowledgement mechanisms are described below because they served as the basis of a simulation experiment discussed later in this paper (Section VI).

One-for-one acknowledgement is permissible within the ISO class 4 transport protocol and is specified within the NBS formal description [4]. Each normal protocol data unit (DT) must be confirmed by an acknowledgement protocol data unit (AK). In addition, DTs received correctly following a DT in error may be retained by the receiver, but must not be confirmed until the erroneous DT has been recovered through retransmission (caused by expiration of a timer). Thus, the DTs are acknowledged in sequence.

TABLE 1
PERFORMANCE OF THE TRANSPORT SIMULATION PROGRAM

ACKNOWLEDGEMENT MECHANISM	ERROR RATE	SIMULATED EVENTS/CPU SECOND	SIMULATED SECONDS/ CPU SECOND	SIMULATED TOTAL BITS/ CPU SECOND	SIMULATED USER BITS/ CPU SECOND
one-for-one VAX-11/780 with fpa	10^{-9}	183	.20	13,952	11,200
	10^{-8}	185	.20	13,952	11,304
	10^{-7}	185	.20	13,952	11,296
	10^{-6}	185	.20	13,952	11,160
	10^{-5}	192	.23	16,074	9,768
	10^{-4}	120	.16	11,100	4,424
	10^{-3}	172	.38	24,515	2,896
one-for-one VAX-11/780 without fpa	10^{-9}	147	.16	11,162	9,024
	10^{-8}	149	.16	11,162	9,120
	10^{-7}	147	.16	11,162	9,000
	10^{-6}	151	.17	11,859	8,840
	10^{-5}	147	.18	12,419	7,304
	10^{-4}	128	.17	11,837	4,744
	10^{-3}	114	.25	16,128	1,704
selective VAX-11/780 without fpa	10^{-9}	117	.13	9,318	7,184
	10^{-8}	118	.13	9,318	7,224
	10^{-7}	118	.13	9,318	7,208
	10^{-6}	116	.13	9,318	7,200
	10^{-5}	106	.12	8,602	6,472
	10^{-4}	89	.10	6,963	5,272
	10^{-3}	111	.29	14,106	3,968

The general case from which the one-for-one scheme derives is the one-for-N acknowledgement mechanism. This is an acknowledgement strategy allowed by the ISO standard transport [12]. In the one-for-N scheme, one AK must be sent for at least every N DTs received. Further, an AK must be sent for the last DT of a user message, i.e., a transport service data unit (TSDU). However, each DT must be acknowledged within some guaranteed maximum time. The rules dealing with erroneous DTs are the same as those used in the one-for-one mechanism.

The selective acknowledgement procedure permits individual confirmation of each DT including those received after erroneous DTs. The AKs may also acknowledge a range of DTs, thus providing a degree of redundancy to protect against damage or loss of AKs. As in the one-for-one mechanism, an AK is generated for each DT received. Although not presently permitted within the ISO transport specification, a selective acknowledgement mechanism may be proposed to ISO as an optional enhancement of the class 4 transport protocol. The specifics of such a proposal are now under study by COMSAT and the NBS [10].

IV. PERFORMANCE METRICS

Our OSI class 4 transport simulation model provides measures of throughput, delay, efficiency, utilization, and retransmissions. An overview of these metrics follows.

A. THROUGHPUT AND DELAY

The model defines the measure of user throughput to be the amount of user information transferred per unit of time as measured at the receiving transport user. The throughput measure is used to determine throughput efficiency, the ratio of user throughput to maximum theoretical throughput as bounded by the speed of the subnetwork access link.

Throughput efficiency is reduced from the ideal (1.0) by such factors as protocol headers, retransmissions, inadequate flow control synchronization, and CPU saturation.

Our simulation measures one-way delay per user message and computes an average per-octet delay per user message to account for variations in delay due to differences in user message sizes. A measure of transport power is also provided as a ratio of throughput efficiency to normalized average per-octet delay.

B. EFFICIENCY AND UTILIZATION

Measures of resource use are reported by our model for the CPU, memory, and communications channel. These measures include protocol efficiency, channel utilization and efficiency, processor utilization, and memory utilization, as described below.

Protocol efficiency is defined as the ratio of the number of user information bits sent to the total number of bits sent. Channel utilization is defined to be the ratio of the number of bits sent on the subnetwork access link per unit time to the maximum number of bits that could be sent on the link per unit time. Deviation from the ideal for this metric is due to idle time on the access link.

Channel efficiency is the ratio of the number of user information bits sent per unit time to the maximum number of user information bits that could be sent per unit time over the subnetwork access link. Two factors cause lowered channel efficiency: 1) protocol overhead (as measured by protocol efficiency) and, 2) the overhead associated with periods of idleness on the subnetwork access link (as measured by channel utilization).

Our model measures memory utilization for the send and receive buffers of each transport module being simulated. The measures provided include a distribution of memory use over time and the average and maximum memory utilizations.

CPU utilization is reported by our model for each simulated processor. CPU utilization is the ratio of seconds that a CPU was in use by the protocol process to the total seconds that the same CPU was available as a resource.

C. RETRANSMISSIONS

Since retransmissions are a major factor in the performance of the ISO class 4 transport protocol, our model has been implemented to provide several specific measures describing the retransmission of data messages. The retransmission measures include the total and maximum number of retransmissions of a data message, a frequency distribution of retransmissions, and the rate of retransmissions per data message.

V. VALIDATION

Validation is a concern to those who rely upon results from a simulation model. We have invested significant resources toward validating our class 4 transport model. In the early stages of design, analytical models [8, 9] provided us with insight into the properties of various aspects of our simulation model. Results were calculated using the analytical models so that later simulation results could be compared with the exact analytical solutions.

As a second step toward validation of our transport model, we had an independent researcher design and implement a simple prototype simulation model that represented several important aspects of the transport protocol not adequately represented by available analytical models. Results from the prototype model were validated where possible with the earlier results from analytical models. We then conducted a series of simulation experiments with the prototype model developing a set of results that could be used to validate a more detailed model as it was completed. The detailed class 4 transport simulation model was then developed and experiments, already run through the prototype model, were repeated with the detailed model.

Having gained some confidence in our model, we are conducting simulation experiments to predict the performance of the transport protocol when operated over satellite channels [10]. The results from each simulation run are examined to verify that the predictions are consistent, understandable, and reasonable. The next step in the validation procedure is to conduct live tests over a satellite link with the class 4 transport protocol, as implemented by COMSAT. The results from the live tests will be compared statistically with the simulation data.

VI. SOME RESULTS

To indicate the utility of our model, we present the results of a simulation experiment comparing one-for-one acknowledgement with a selective acknowledgement procedure. The operation of the one-for-one acknowledgement mechanism is as specified in the international standard transport protocol specification [4].

Selective acknowledgement, as used in the experiment, was designed, formally specified, and implemented using automated tools previously developed at the NBS [7]. A formal specification for selective acknowledgement was produced by modifying the NBS formal description of class 4 transport. The resulting protocol extends the ISO class 4 transport so that the AK transport protocol data unit (TPDU) contains two sequence numbers that define a range of DT TPDUs to be acknowledged. The operation of the selective mechanism was described above (Section III. C.). A more complete specification is contained in a formal description available from the authors.

A. THE EXPERIMENT

The simulation experiment reported here involved sending continuously queued data in one direction on a single transport connection across a satellite communications channel. The transport protocol implementations simulated were identical dual-processor configurations of a COMSAT Laboratories network interface processor (NIP).

The experiment was conducted twice with the same set of parameters, i.e., only the acknowledgement mechanism (one-for-one and selective) was changed for each trial. The significant experiment parameters are given in Table 2. Several earlier experiments were conducted to select appropriate timer values, TPDU size, network queue size, transport window size, and buffer memory. Although interesting in themselves, the earlier experiments are not reported here.

B. PERFORMANCE IMPROVEMENTS

The results of the experiment indicate that the selective acknowledgement mechanism provides more stable performance (for the measures of throughput efficiency, protocol efficiency, and delay) over a range of deteriorating error rates. These results are shown in Figures 3, 4, and 5.

Figure 3 illustrates the variation in throughput efficiency as the error rate on a channel deteriorates from 10^{-9} to 10^{-3}. The throughput efficiency of the selective acknowledgement mechanism drops only when the error rate approaches 10^{-3}, while the throughput efficiency of the one-for-one scheme declines much earlier, as the error rate exceeds 10^{-6}.

Figure 4 depicts the trend in protocol efficiency over the range of error rates of interest. For the error rates 10^{-9}, 10^{-8}, and 10^{-7}, the protocol efficiency of the one-for-one acknowledgement mechanism is about 2% greater than that of the selective acknowledgement mechanism, since each AK TPDU for selective acknowledgement is 4 bytes larger.

The protocol efficiency of the selective acknowledgement mechanism falls sharply as the error rate approaches 10^{-3}. The protocol efficiency of the one-for-one acknowledgement mechanism drops much earlier, as the error rate exceeds 10^{-6}.

Table 2

Acknowledgement Experiment Parameters

Class	Parameter	Value(s)
Network Characteristics	Link Speed	64,000 bps
	Propagation Delay	270 ms
	Error Rate	10^{-3} to 10^{-9}
	Queue Size	4 Messages
Transport Characteristics	Receive Memory	32K bytes
	Send Memory	32K bytes
	Window Size	20K bytes
	Maximum DT TPDU Size	128 bytes
	Retransmission Timer	650 ms
	Window Timer	2000 ms
Data Characteristics	TSDU size	10K bytes
	Number of TSDUs	20 (5 at 10^{-3})
	Arrival Distribution	Continuously Queued
	Directionality	Uni-directional

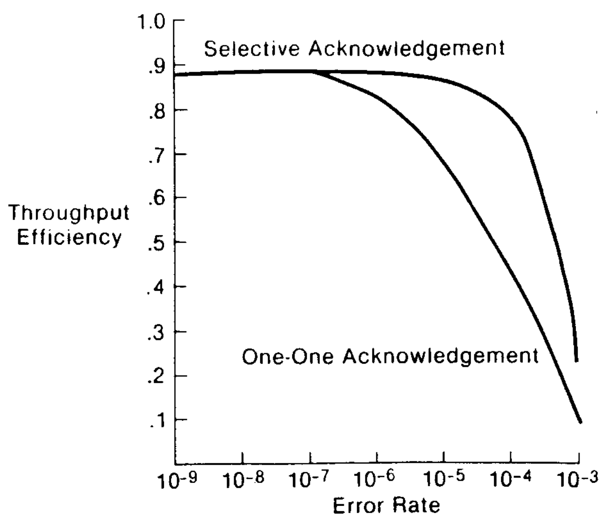

Figure 3. Throughput Efficiency vs Error Rate

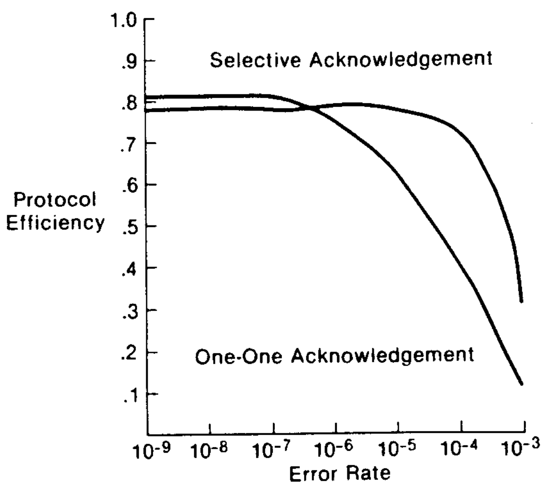

Figure 4. Protocol Efficiency vs Error Rate

Figure 5 shows the trend in average delay as the error rate deteriorates. The results for delay support those shown for throughput and protocol efficiencies. The one-for-one acknowledgement strategy exhibits increases in delay beyond the 10^{-6} error rate while selective acknowledgement provides stable delay through the 10^{-4} error rate.

Assuming that one would choose not to use a network service where error rates of worse than 10^{-5} were experienced, where can the selective acknowledgement mechanism provide substantial benefits? The appropriate environments are those where the error rate can vary drastically over time. For example, a satellite link may experience variations in the error rate due to weather changes and other changing sources of interference. In a military environment where line of sight microwave and tropospheric scatter are used, the use of selective acknowledgement mechanisms may give better performance than one-for-one acknowledgement strategies.

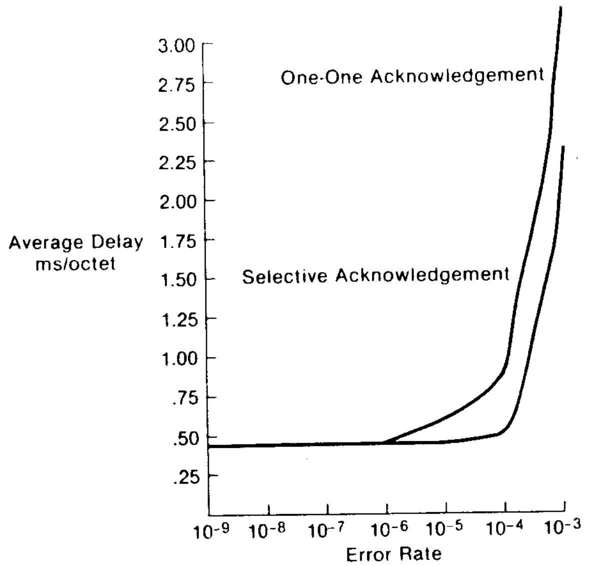

Figure 5. Delay vs Error Rate

C. COST OF IMPROVEMENTS

As one should expect, performance improvements are not free. In the case of selective acknowledgement mechanisms, we have already shown that in error-free environments the protocol efficiency is not quite as good as that of one-for-one acknowledgement (due to the requirement for four extra bytes per AK). A larger cost associated with the selective acknowledgement mechanism, as illustrated in Figures 6 and 7, is related to the fact that a window of twice the normal size must be granted to permit the continuous flow of original transmissions while awaiting, in case of an early transmission error, retransmission of the first original message in the window.

Figure 6 depicts the average utilization of receiver memory as the error rate varies from 10^{-9} to 10^{-3}. As the error rate reaches and surpasses 10^{-5}, the average receive memory utilization for the selective acknowledgement mechanism exceeds that of one-for-one acknowledgement. This result shows that the improvements in throughput efficiency, protocol efficiency, and delay come at the expense of memory utilization at the receiver.

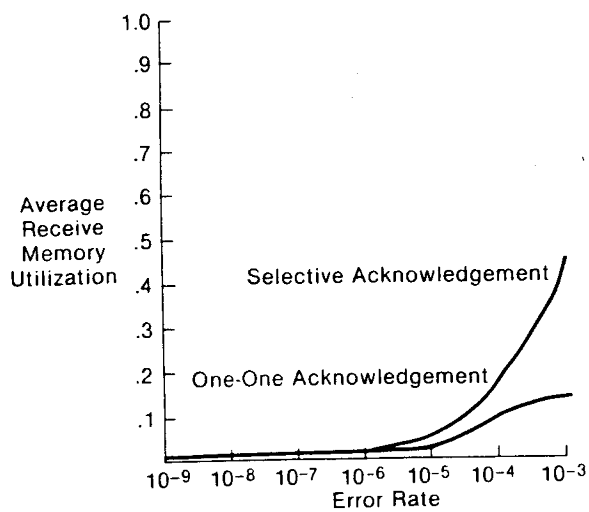

Figure 6. Average Receive Memory Utilization vs Error Rate

The cost of these improvements, in terms of average memory utilization, may be considered minor. However, the cost in terms of maximum memory utilization is much greater, as shown in Figure 7. When using selective acknowledgement, average memory utilization as high as 45% was observed, but the maximum memory utilization was as high as 93%. Therefore, to obtain the performance improvements of selective acknowledgement, more memory must be available, although for much of the time a substantial portion of the available memory will be unused.

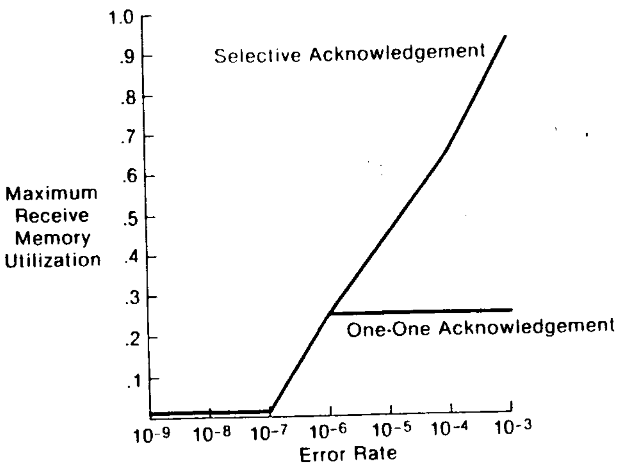

Figure 7. Maximum Receive Memory Utilization vs Error Rate

The increase in memory utilization is due to the fact that fewer retransmissions (and therefore more original transmissions) arrive at the receiver per unit of time. This is due to the large window and the selective acknowledgements. The arriving transmissions may not be forwarded on to the receiving user except as a sequenced stream of data. Therefore, the receiving transport must buffer the correct transmissions, pending receipt of retransmissions for messages that were damaged during transmission. Many more original messages can be sent in a specified period of time when selective acknowledgement is used and, thus, many more messages must be buffered by the receiver.

VII. FUTURE WORK

We will continue to work with transport protocol simulation in two areas. First, we will perform further experiments including: 1) evaluating additional acknowledgement schemes, 2) comparing mechanisms for providing the ISO transport expedited service, 3) examining performance for multiplexed and full-duplex transport connections, 4) predicting the effects of various buffer management techniques, and 5) considering mechanisms for monitoring and enforcing quality of service on a set of transport connections. These planned experiments lead naturally to consideration of various communications subnetwork technologies and user traffic patterns. Thus, several requirements exist for changes to our simulation model. Meeting the new requirements is our second area of future work.

We plan to improve the model in three ways. First, we will increase the flexibility available for specifying transport user traffic. This improvement requires traffic generation functions for several statistical distributions to simulate transport user traffic interarrival times and message lengths. Second, we will extend our network model to a more general specification of network properties, allowing representation of LANs, concatenated networks, and public data networks, in addition to satellite channels. Finally, we plan to change the simulated transport scheduling mechanism to permit the assignment of priorities to specific transport connections. The present scheduling mechanism among transport connections is first come, first served. Improvements to the simulated scheduling mechanism are necessary in order to permit investigation of strategies for meeting transport user requirements for throughput and delay (i.e., quality of service).

VIII. CONCLUSION

The ICST has extended the scope of its computer networking program to include protocol performance research. This paper has described a simulation model of the ISO class 4 transport protocol, a first step in the ICST program of protocol performance research. The power and flexibility of the model was demonstrated in two areas. First, the simulation model implements the data transfer phase of class 4 transport and several alternative protocol mechanisms in sufficient detail to allow for simulation of a variety of implementations. This detail contributes significantly to realistic results. Second, the metrics produced by the model cover many facets of protocol performance. Taken together, these metrics can lead to interesting conclusions about characteristics of protocol performance, such as was demonstrated with the two acknowledgement schemes described.

The chief benefit of employing this model is the ability to investigate a more extensive set of issues than would be possible with live experiments or analytical models. Simulation modeling increases our insight into the behavior of class 4 transport protocol implementations and allows us to make more effective use of our project resources.

IX. ACKNOWLEDGEMENTS

Marnie Wheatley and Kevin Mills designed, implemented, and validated the simulation model of class 4 transport. Richard Colella produced the formal specifications of the alternative protocol algorithms. Kevin Mills defined the performance metrics and conducted the acknowledgement experiment reported in this paper. Anastase Nakassis designed and implemented the simple prototype simulation and designed the module clocking control algorithm used in the class 4 transport model.

X. REFERENCES

[1] J. Stephen Nightingale. "Protocol Testing Using a Reference Implementation." Proceedings of the IFIP WG 6.1 Second International Workshop on Protocol Specification, Testing, and Verification, May, 1982.

[2] Kevin L. Mills. "Testing OSI Protocols: NBS Advances the State of the Art." Data Communications, March 1984.

[3] Richard Colella. "The NBS Performance Program for OSI Protocols." This document is in preparation.

[4] "Specification of a Transport Protocol for Computer Communications, Volume 3: Class 4 Protocol." Report No. ICST/ HLNP-83-3, National Bureau of Standards, February, 1983.

[5] "Specification of a Transport Protocol for Computer Communications, Volume 4: Service Specifications." Report No. ICST/HLNP-83-4, National Bureau of Standards, January, 1983.

[6] Charles H. Sauer and Edward A. MacNair. Simulation of Computer Communication Systems. Prentice-Hall, 1983.

[7] "A Formal Specification Technique and Implementation Method for Protocols." Report No. ICST/HLNP-81-15, National Bureau of Standards, July, 1981.

[8] James Martin. Communications Satellite Systems. Prentice-Hall, 1978.

[9] Dattakumar M. Chitre. "Analysis of Throughput Efficiency and Delay in ARQ Systems." COMSAT Technical Review, Volume 11, Number 2, Fall 1981.

[10] Marnie Wheatley, Richard Colella. "COMSAT/NBS Experiment Plan for Transport Protocol." Report No. NBSIR 85-3141, National Bureau of Standards May 1985.

[11] "Information Processing Systems - Open Systems Interconnection Transport Service Definition." International Standards Organization, IS 8072, 1984.

[12] "Information Processing Systems - Open Systems Interconnection Connection - Oriented Transport Protocol." International Standards Organization, IS 8073, 1984.

[13] "Specification of a Transport Protocol for Computer Communications Volume 1: Overview and Services." Report No. ICST/HLNP-83-1, National Bureau of Standards, January, 1983.

[14] "Specification of a Transport Protocol for Computer Communications Volume 5: Guidance for the Implementor." Report No. ICST/HLNP-835, National Bureau of Standards, January, 1983.